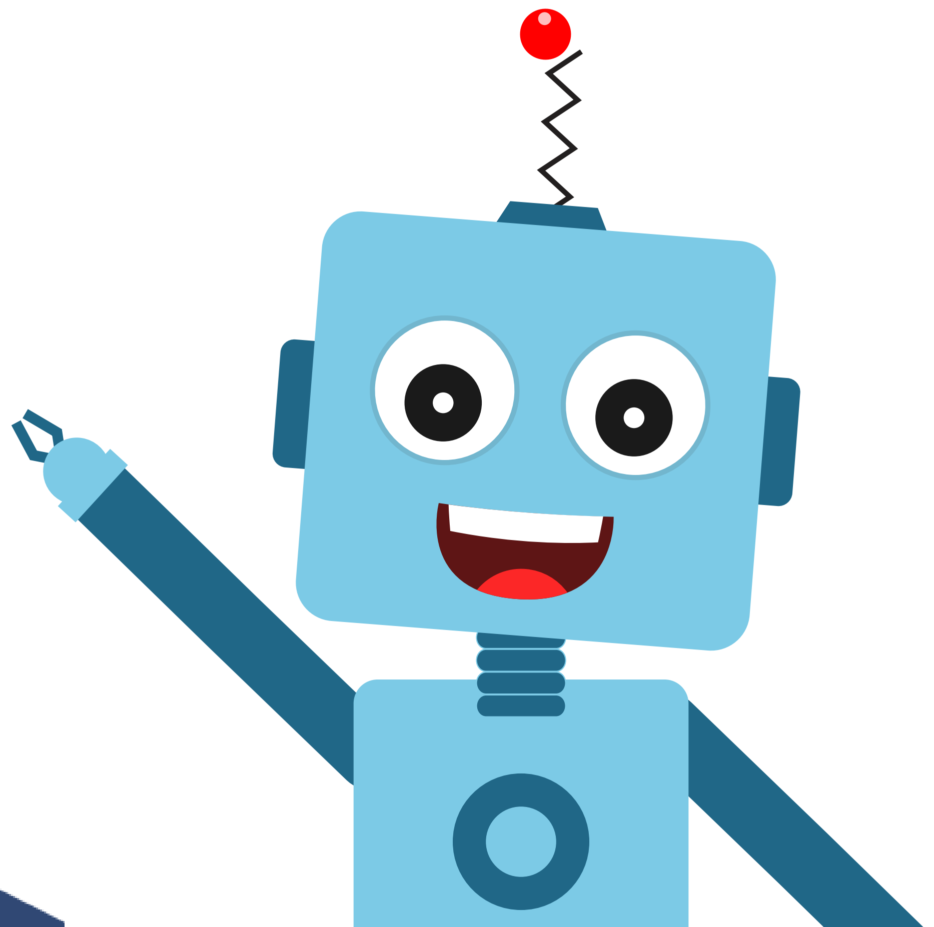
POWERED BY STEMpedia

Lists in Python

Session 10



Topics covered

POWERED BY STEMpedia

1. Basics of list

2. Activity :
    1. Manipulating list

POWERED BY STEMpedia

Basics of List

A list in Python is an ordered collection of objects, which can be of any type such as integers, strings, dictionaries, and even other lists

Introduction

- Like a String, a list also is a sequence data type. It is an ordered set of values enclosed in square brackets [].
- Values in the list can be modified, i.e. it is mutable.
- As it is a set of values, we can use the index in square brackets [] to identify a value belonging to it.

Let’s look at some example of simple list:

1. L1 = [1, 2, 3, 4] # list of 4 integer elements.
2. L2 = [“Delhi”, “Chennai”, “Mumbai”] #list of 3 string elements.
3. L3 = [ ] # empty list i.e. list with no element
4. L4 = [“abc”, 10, 20] # list with different types of elements

Elements of the List

POWERED BY STEMpedia

- For accessing an element of the list, indexing is used. Its syntax is:
- Variable name [index] (the variable name is the name of the list)
- The positive value of the index means counting forward from beginning of the list .
- A negative value means counting backward from the end of the list.

Example:

```
L1 = [1, 2, 3, 4]
print(L1[2])
```

```
>> 3
```

Here, 3rd element of the list (accessed using index value 2) is 3.

Changing the List

- To change the value of an element of the list, we access the element & assign the new value.

```
L1 = [1, 2, 3, 4]
L1[2] = 6
print(L1)
```

```
>> [1,  2, 6, 4]
```

Here, 3rd element of the list (accessed using index value 2) is given a new value, so instead of 3, it will be 6.

Creating a list

POWERED BY STEMpedia

The list can be created in many ways:

- By enclosing elements in [ ], as we have done in the above examples.

```
L1 = [1, 2, 3, 4]
```

- Using other Lists

```
L1 = [1, 2, 3, 4]
L2 = L1[:]
print(L2)
```

```
>> [1, 2, 3, 4]
```

Here L5 is created as a copy of L1.

Creating a list

POWERED BY STEMpedia

- List comprehension

```
n = 5
L4 = range(n)
print([*L4])
```

```
>> [1, 2, 3, 4, 5]
```

```
A = [3, 4, 5]
B = [value*3 for value in A]
print(B)
```

```
>> [9, 12, 15]
```

Here B will be created with the help of A and every element will be thrice of the element of A.

Subscript

An individual character in a string is accessed using a subscript (index). The subscript should always be an integer (positive or negative). A subscript starts from 0.

1. To access the first character of the string:

```
print(message[0])
```

2. To access the fourth character of the string:

```
print(message[3])
```

3. To access the last character of the string:

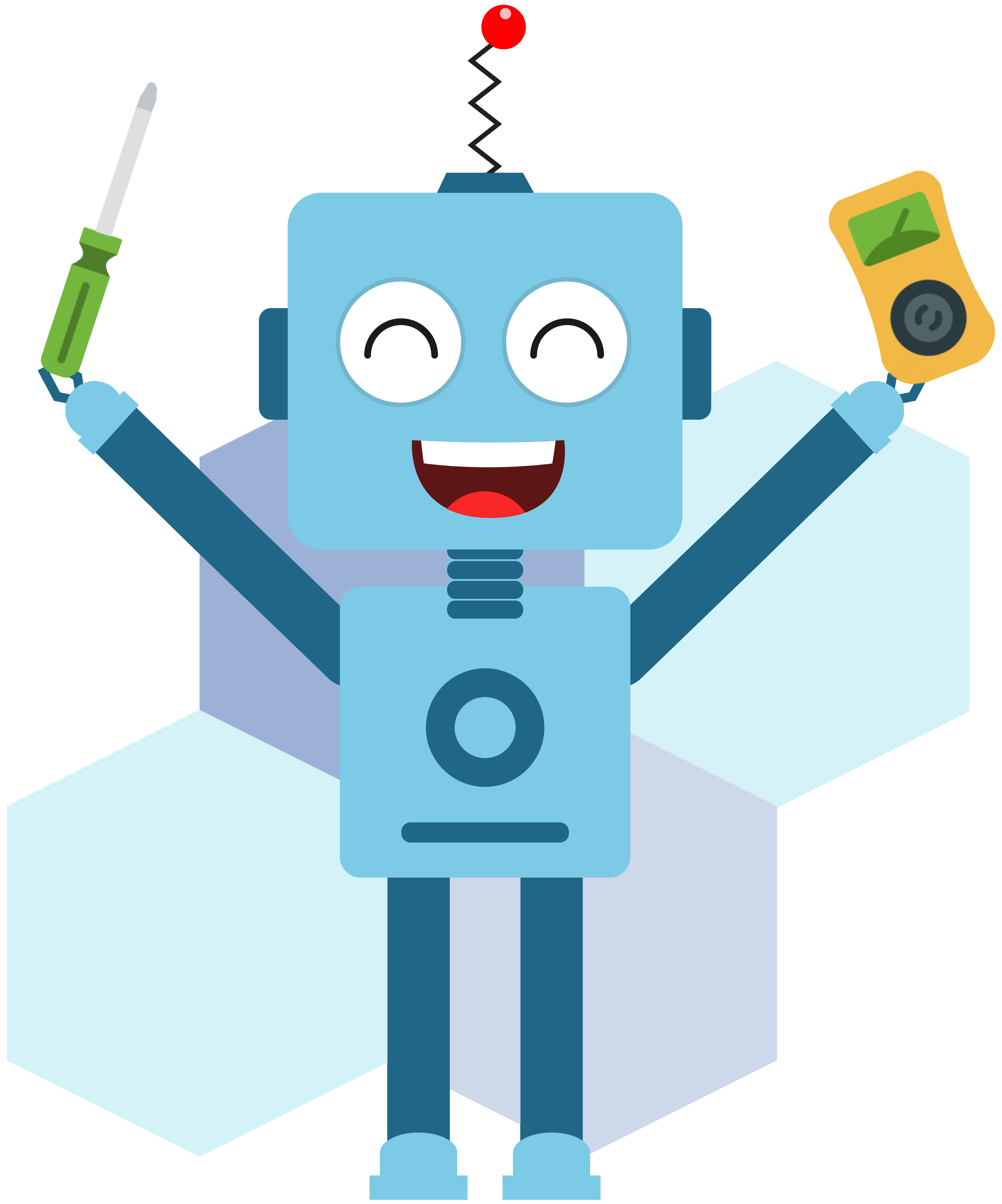
```
print(message[-1])
```

ACTIVITY

POWERED BY STEMpedia

Manipulating a List

For accessing an element, we use index and we have already seen examples doing so. To access an element of a list containing another list, we use pair of indexes.



Accessing an element of the list

- For accessing an element, we use index and we have already seen examples doing so.

- To access an element of a list containing another list, we use pair of indexes.

  Let's understand it with the following List:

  L = [1, [1, 2, 3], "Hi", 5]

List Slices

- The slice operator works on the list also. We know that a **slice of a list is its sub-list**. We use the [n:m] operator to create a list slice.

```
print(L[0])
>> 1
```

List Slices

POWERED BY STEMpedia

- As the 2nd element of this list is a list. To access a value from this sub-list, we will use:

```
print(L[1][1])
```
```
>> 2
```

- L[n:m] will return the part of the list from the nth element to the mth element, **including the first element but excluding the last element.** So the resultant list will have “m-n” elements in it.

- Example: L[1:2] will have m-n = 2-1 = 1 element in it

```
print(L[1:2])
```
```
>>[[1, 2, 3]]
```

Syntax of slicing

- **sequence = L [start: stop: step],** where start, stop & step- all three are optional.
- If you omit the first index(**start**), the slice starts from "0", omitting the **stop** will take it to the end. If you omit the **step**, the default value of the step will be "1".

Example: L = [10, 20, 30, 40, 50, 60]

1. `print(L[::2])` will produce a list with alternate elements.

```
>> [10, 30, 50]
```

2. `print(L[4:])` will produce a list containing all the elements from 5th position till end.

```
>> 50, 60
```

3. `print(L[-1])` "-1" refers to last elements of list.

```
>> 60
```

Traversing a List

POWERED BY STEMpedia

- Using while loop:

```
L = [1, 2, 3, 4, 5]
index = 0
while index < 5:
print(L[index])
index = index + 1
```

```
>> 1
>> 2
>> 3
>> 4
>> 5
```

Traversing a List

POWERED BY STEMpedia

- Using for loop:

```
L = [1, 2, 3, 4, 5]
for i in L:
print(i)
```

```
>> 1
>> 2
>> 3
>> 4
>> 5
```

VOWELS IN STRING

In This activity we write a code uses a for loop to iterate through each character in the input string and checks if the character is a vowel. If a vowel is found, the count variable is incremented by 1. Finally, the code outputs the total count of vowels found in the input string.

Vowels in string

- First we asks the user to enter a string and stores the input in a variable called str.
- Then we initialize a variable called count to 0. This variable will be used to keep track of the number of vowels found in the string.

```
#Taking input from the user
str = input("Enter a string: ")
#initialize a count variable
count = 0
```

```
>> 1
>> 2
>> 3
>> 4
>> 5
```

Vowels in string

POWERED BY STEMpedia

- Furthermore, we start a for loop that iterates through each character in the string using the range function and the len function to obtain the length of the string.

```
for i in range(len(str)):
```

- Then we check if the current character in the string (accessed using the index i) is a vowel. The str[i].upper() expression converts the current character to uppercase so that we can check for both lowercase and uppercase vowels. If the character is a vowel, the if statement evaluates to True.

```
#check if any part of the string exists in the list
of vowels
    if str[i].upper() in ['A', 'E', 'I', 'O', 'U']:
```

Vowels in string

- Then we increment the count variable by 1 if the current character is a vowel.

```
count += 1
```

- Finally, it gives output the total count of vowels found in the string using a formatted string.

```
print("The number of vowels is: {}".format(count))
```

Vowels in string (Final code)

```
#Taking input from the user
str = input("Enter a string: ")
#initialize a count variable
count = 0
for i in range(len(str)):
#check if any part of the string exists in the list of vowels
    if str[i].upper() in ['A', 'E', 'I', 'O', 'U']:
        count += 1
print("The number of vowels is: {}".format(count))
```

POWERED BY STEMpedia

www.ai.thestempedia.com